Predicción de la producción de energía de una planta de poder de ciclo combinado usando herramientas de AutoML

Gabriel Alejandro Aguilar Farrera

***Abstract*—Este estudio se enfoca en predecir la producción neta de energía eléctrica por hora de una planta de poder de ciclo combinado. Se exploran diversas técnicas de regresión, incluyendo regresión Ridge y Lasso, junto con herramientas de automatización como AutoKeras y PyCaret. El objetivo es identificar el mejor modelo de regresión para este problema específico. Además, se utilizan métodos de búsqueda de hiperparámetros como GridSearchCV, RandomizedSearchCV y HalvingSearchCV para encontrar los valores óptimos de lambda en los casos de Ridge y Lasso, maximizando así la capacidad predictiva del modelo. Este informe resume los procedimientos utilizados, los resultados obtenidos y las conclusiones derivadas de la experimentación computacional.**

***Index Terms*—Regresión Ridge y Lasso, AutoKeras y PyCaret.**

I. Introducción

- **Contexto general del problema:** La calidad de los modelos de regresión impacta significativamente en diversas áreas. Estos modelos son fundamentales para predecir y entender relaciones entre variables. La capacidad de prever eventos futuros con precisión es crucial para la planificación estratégica. Los modelos de regresión bien ajustados permiten proyectar tendencias, analizar riesgos y tomar medidas preventivas. Por ejemplo, en el ámbito empresarial, estos modelos respaldan estrategias de marketing basadas en análisis predictivos de ventas. En la investigación científica, los modelos de regresión son vitales para comprender la relación entre variables y fenómenos complejos. Ayudan a identificar factores críticos, influencias y patrones ocultos en datos experimentales, facilitando descubrimientos significativos y avances científicos. En resumen, desarrollar modelos de regresión precisos y confiables es esencial para la toma de decisiones informadas, la planificación estratégica y la investigación científica, impactando directamente en el progreso y desarrollo en diversas áreas de estudio.
- **Contexto particular del problema:** En el ámbito de las plantas de energía, predecir la producción eléctrica con precisión es fundamental para optimizar la eficiencia y reducir costos. La regresión se convierte en una herramienta esencial para este propósito, ya que permite modelar y comprender cómo las variables ambientales, como temperatura, presión y humedad, influyen en la generación de energía. Por ejemplo, en una planta de energía de ciclo combinado, entender cómo la temperatura ambiente afecta la eficiencia de las turbinas de gas y vapor es crucial. Un aumento en la temperatura puede disminuir la densidad del aire, lo que afecta negativamente el rendimiento de las turbinas. Esto puede resultar en una producción eléctrica menor de lo esperado, lo que a su vez impacta en la eficiencia global de la planta y, potencialmente, en los costos operativos. Mediante modelos de regresión como Ridge y Lasso, podemos cuantificar estas relaciones y construir predicciones precisas sobre la producción de energía en función de las condiciones ambientales. Esto permite a los ingenieros y gerentes de planta anticipar cambios en la generación de energía según las variaciones en las variables ambientales, facilitando la toma de decisiones para mantener un rendimiento óptimo y económico de la planta.
- **Descripción general de la propuesta de solución:** En este trabajo, nos enfocaremos en predecir la producción de energía de una planta de poder de ciclo combinado. Utilizaremos modelos avanzados de regresión como Ridge y Lasso, además de herramientas de AutoML como AutoKeras y PyCaret. El objetivo principal es identificar el modelo de regresión óptimo para este caso, evaluando su rendimiento para maximizar la precisión en la predicción de la producción de energía. También aplicaremos métodos de búsqueda de hiperparámetros como GridSearchCV, RandomizedSearchCV y HalvingSearchCV para afinar los modelos de Ridge y Lasso y garantizar su capacidad predictiva óptima.
- **Descripción general de los principales resultados:** Durante la experimentación computacional, se logró identificar el modelo de regresión óptimo para predecir la producción de energía en la planta de poder de ciclo combinado. Se encontró que el modelo Extreme Gradient Boosting ofrecía el rendimiento más sólido y preciso en términos del R^2. Además, los métodos de búsqueda de hiperparámetros demostraron ser efectivos al ajustar los modelos de Ridge y Lasso, mejorando su capacidad predictiva y proporcionando valores óptimos de lambda. Estos resultados no solo validan la utilidad de los modelos de regresión en este contexto, sino que también destacan la eficacia de las herramientas de AutoML y la importancia de la optimización de hiperparámetros

para lograr una mayor precisión en las predicciones.

- **Organización del documento:** En la sección de Materiales y Métodos se da una breve descripción de los modelos de regresión Ridge y Lasso, así como la importancia de elegir un buen valor de λ. De la misma forma se da un énfasis en la importancia del uso y conocimiento de las herramientas de AutoML disponibles hoy en día para la elección del modelo o pipeline óptimo, finalmente en esta sección se explican los datos y en qué consiste el funcionamiento de una planta de poder de ciclo combinado. En la sección de Experimentos y Resultados se detallan los resultados obtenidos y se presentan comparaciones entre ellas. Finalmente, en la sección de Conclusión se mencionan las observaciones más relevantes del trabajo.

II. Materiales y Métodos

- **Descripción de los métodos usados:** Las estrategias utilizadas para encontrar el mejor modelo de predicción son la regresión Ridge, regresión Lasso y métodos de búsqueda de hiperparámetros. Así mismo, herramientas de automatización como AutoKeras y PyCaret. A continuación se da una breve descripción de todas estas herramientas y métodos.

La **Regresión Ridge** es muy similar a mínimos cuadrados, excepto que los coeficientes se estiman minimizando una expresión ligeramente diferente. Las estimaciones de los coeficientes de la regresión ridge $\hat{\beta}_R$ son los valores que minimizan

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = RSS + \lambda\sum_{j=1}^{p}\beta_j^2 \tag{1}$$

donde$\lambda \geq 0$ es un parámetro de ajuste, que se determina por separado. Al igual que mínimos cuadrados, la regresión Ridge [1] busca las estimaciones de los coeficientes que ajusten bien a los datos haciendo que el RSS sea pequeño. El segundo término, $\lambda\sum_{j=1}^{p}\beta_j^2$ se denomina *penalización por contracción* y es pequeño cuando $\beta_1, ..., \beta_p$ con cercanos a cero. Por lo que tiene el efecto de reducir las estimaciones de β_j a cero. El parámetro de ajuste λ sirve para controlar el impacto relativo de estos dos términos en las estimaciones del coeficiente de regresión. Cuando $\lambda = 0$, el término de la penalización no tiene efecto, y la regresión ridge producirá estimaciones de mínimos cuadrados. Por otro lado, cuando $\lambda \to \infty$ el impacto de la penalización por contracción crece y las estimaciones del coeficiente de regresión ridge se aproximarán a cero. A diferencia de los mínimos cuadrados que generan un solo conjunto de estimaciones de coeficientes, la regresión ridge producirá diferentes conjuntos de estimaciones de coeficientes $\hat{\beta}_\lambda^R$, para cada valor de λ. Notemos que la penalización por contracción se aplica a $\beta_1, ..., \beta_p$, pero no al intercepto β_0. Finalmente es importante decir que las estimaciones de los coeficientes de ridge no son invariantes a la escala de medición, por lo que es mejor aplicar regresión ridge después de estandarizar las variables predictoras.

Al igual que con ridge, la **Regresión lasso** reduce las estimaciones de los coeficientes hacia cero. Los coeficientes Lasso $\hat{\beta}_\lambda^L$, se obtienen minimizando la expresión:

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda\sum_{j=1}^{p}|\beta_j| = RSS + \lambda\sum_{j=1}^{p}|\beta_j| \tag{2}$$

La expresión (2) es muy similar a (1) sólo que Lasso usa la norma l_1 en la penalización, en lugar de la norma l_2 usada en Ridge. En el caso de Lasso [1], la penalización l_1 tiene el efecto de forzar algunas de las estimaciones de coeficientes para que sean exactamente cero, cuando el parámetro de ajuste λ es suficientemente grande, por lo que Lasso (a diferencia de Ridge) si realiza una selección de variables. Así, los modelos generados a partir de Lasso generalmente son mucho más fáciles de interpretar que los producidos por Ridge. Cuando $\lambda = 0$, los coeficientes Lasso son simplemente los obtenidos con mínimos cuadrados. Cuando λ es suficientemente grande todos los coeficientes estimados son iguales a cero. Para la implementación de Ridge y Lasso el método usual para seleccionar el valor apropiado para λ es validación cruzada: Elegimos un conjunto de valores para λ y calculamos el error de validación cruzada para cada valor de λ. En este proyecto vamos a ocupar los métodos GridSearchCV (técnica de búsqueda exhaustiva de hiperparámetros), RandomizedSearchCV (técnica que realiza muestreos aleatorios en lugar de explorar todos los valores en una cuadrícula) y HalvingSearchCV (técnica que implica una estrategia de reducción progresiva) para seleccionar el valor de λ apropiado, así mismo se estudiarán sus tiempos de ejecución y MSE en el conjunto de train y test para determinar cuál de estos métodos es el más apropiado y si en verdad existe una mejora significativa entre un método y otro en cuestiones del valor de MSE y tiempo computacional.

El Aprendizaje Automático Automatizado (AutoML) es una forma computarizada de determinar la mejor combinación de preparación de datos, modelo e hiperparámetros para una tarea de modelado predictivo. El modelo de AutoML tiene como objetivo automatizar todas las acciones que requieren más tiempo, como la selección de algoritmos, la escritura de código, la creación de pipelines, etc. En resumen, AutoML puede ayudar a los profesionales del aprendizaje automático a realizar trabajos de modelado predictivo de manera rápida y eficiente con una entrada mínima. Hay varias herramientas de código abierto de AutoML disponibles en el mercado que aceleran los procesos de Machine Learning, y dos de esas herramientas son '**AutoKeras**' y '**PyCaret**'. Mientras que AutoKeras se centra en la automatización del diseño y la optimización de arquitecturas

de redes neuronales, PyCaret ofrece una solución integral para simplificar y agilizar todo el proceso de construcción, comparación y selección de modelos de aprendizaje automático.

- **Descripción de los datos utilizados:** El conjunto de datos con el que se trabajo se titula 'Combined Cycle Power Plant' [2] y se puede descargar de forma gratuita desde la página de Machine Learning Repository. El conjunto de datos contiene 9568 puntos de datos recopilados de una planta de energía de ciclo combinado durante 6 años (2006-2011), cuando la planta de energía estaba configurada para funcionar a carga completa. Las características consisten en variables ambientales promedio por hora: Temperatura (T), Presión Ambiental (AP), Humedad Relativa (RH) y Vacío de Escape (V) para predecir la salida de energía eléctrica neta por hora (EP) de la planta. Las características consisten en variables ambientales promedio por hora:

  Temperatura (T) en el rango de 1.81°C a 37.11°C.

  Presión Ambiental (AP) en el rango de 992.89-1033.30 milibares.

  Humedad Relativa (RH) en el rango de 25.56% a 100.16%.

  Vacío de Escape (V) en el rango de 25.36-81.56 cm Hg.

  Salida de energía eléctrica neta por hora (EP) de 420.26-495.76 MW.

  Los promedios se toman de varios sensores ubicados alrededor de la planta que registran las variables ambientales cada segundo. Las variables se presentan sin normalización. Una planta de energía de ciclo combinado (CCPP) está compuesta por turbinas de gas (GT), turbinas de vapor (ST) y generadores de vapor de recuperación de calor. En un CCPP, la electricidad se genera mediante turbinas de gas y vapor, que están combinadas en un ciclo y se transfieren de una turbina a otra. Mientras que el Vacío se recopila y tiene efecto sobre la Turbina de Vapor, las otras tres variables ambientales afectan al rendimiento del GT. En las Figuras (1) y (2) se muestra con más detalle el funcionamiento de una planta de energía de ciclo combinado.

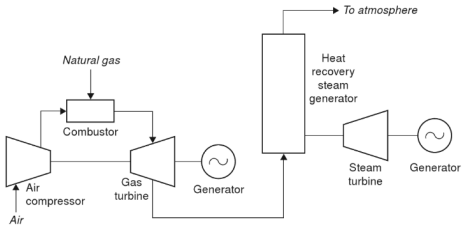

Fig. 1: Esquema de trabajo de una CCPP.

Fig. 2: Visualización de una CCPP.

III. Experimentos y Resultados

- **Descripción del diseño experimental:** Nuestro análisis se enfoca en una evaluación exhaustiva y comparativa de los modelos de regresión Ridge y Lasso, junto con herramientas de AutoML como AutoKeras (desde una perspectiva de deep learning) y PyCaret (desde una perspectiva de machine learning). A continuación se detalla el paso a paso de la experimentación computacional que se realizó para encontrar el pipeline óptimo para la predicción de energía de una CCPP. Primero, detallamos el enfoque bajo los modelos de regresión ridge y lasso:

  Análisis de correlación lineal: Iniciamos analizando la correlación existente entre los regresores (T, AP, RH y V) y la variable a predecir (EP).

  División de datos: Dividimos nuestro conjunto de datos en una proporción de entrenamiento y prueba, utilizando un 80 % para entrenamiento y un 20 % para prueba. Este enfoque estratégico nos permitió realizar el entrenamiento de nuestros modelos en un conjunto de datos independiente, reservando el conjunto de prueba para la evaluación final del rendimiento de los modelos.

  Escalado de los datos: Escalamos los datos y evaluamos su desempeño en los modelos de regresión. En este marco de trabajo consideramos dos escalamientos diferentes: normalización (3) y estandarización (4) de los datos.

$$X_{normalized} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{3}$$

$$X_{standardized} = \frac{X - \mu}{\sigma} \tag{4}$$

  Gráfica de los coeficientes Ridge y Lasso en función de λ: Esto nos ayuda a conocer que valores de λ nos dan como resultado la solución de mínimos cuadrados o similar. De esta forma podemos explorar los valores de λ que si realizan contracción de los valores β_j ya sea en ridge o lasso.

  Rejilla de valores posibles para λ (penalización por contracción): Definimos un espacio de valores posibles para λ y con los métodos de búsqueda de hiperparámetros antes mencionados seleccionamos el valor óptimo para los modelos de Ridge y Lasso.

  Creación del mejor modelo Ridge y Lasso: Se crean los modelos Ridge y Lasso con los valores de λ escogidos.

Evaluación de los modelos: Se evalúan los modelos en los datos de entrenamiento y prueba. Las métricas que se utilizan para hacer la evaluación son el R^2 y el *MSE*. Al igual que en el caso anterior, bajo el enfoque de AutoKeras también le damos los datos normalizados y estandarizados como input al objeto encargado de hacer la búsqueda de arquitecturas neuronales. AutoKeras tiene un par parámetros que el usuario puede configurar, entre ellos están *max_trials* el cual indica el número máximo de arquitecturas neuronales y/o combinaciones de hiperparámetros a probar durante la búsqueda y *epochs* el cual indica cuantas épocas como máximo se va a entrenar cada arquitectura. Finalmente, se elige la arquitectura que tenga los valores más bajos de MSE. Por otro lado, bajo el enfoque de PyCaret se realiza una comparación de todos los pipenlines con los que cuenta ésta herramienta y se realiza una comparación entre el mejor modelo de PyCaret con los que hemos implementado en AutoKeras y en los modelos de regresión Ridge y Lasso.

- **Descripción de los análisis a realizar:**

  Bajo el enfoque de regresión ridge y lasso, se analiza los valores posibles en que λ proporciona una solución diferente a la de mínimos cuadrados y si esos resultados son mejores que los dados por mínimos cuadrados.

  Hacer una comparativa entre la complejidad e interpretabilidad de los modelos aquí analizados, así como sus métricas (R^2 y MSE) en los conjuntos de entrenamiento y prueba.

- **Resultados y discusión:** A continuación se presentan los resultados obtenidos. Como primer punto se muestra la correlación existente entre los regresores y la variable a predecir (ver Fig. 3).

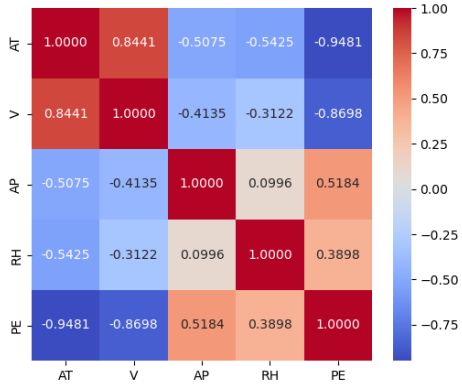

Fig. 3: Correlación lineal entre los regresores (T, AP, RH y V) y la variable a predecir (EP).

En la Fig. (3) se puede observar que la temperatura (T) y el vacío de escape tienen una correlación lineal negativa muy fuerte con la salida de energía eléctrica neta por hora (EP), mientras que el resto de variables (AP y RH) tienen una correlación lineal baja con la EP (menor a 0.6). Sin embargo, ya que tenemos tan solo 4 regresores no haremos exclusión de ninguno de ellos. A continuación se muestran los valores de los coeficientes de Ridge y Lasso en función del valor de λ con los datos normalizados (ver Fig. 4) y con los datos estandarizados (ver Fig. 6).

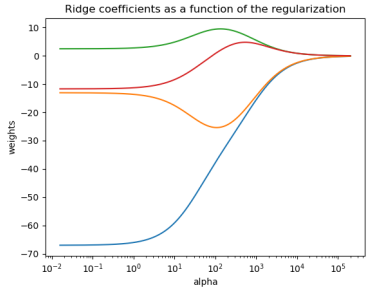

Fig. 4: **Datos normalizados:** Coeficientes de Ridge en función del valor de λ.

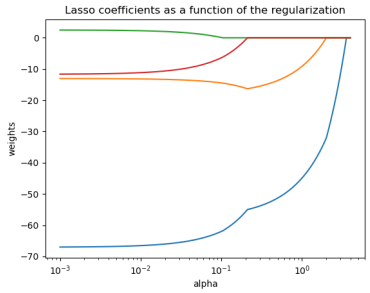

Fig. 5: **Datos normalizados:** Coeficientes de Lasso en función del valor de λ.

En la Fig.(4) se muestra la ventana de valores que se explora en los métodos de GridSearchCV, RandomizedSearchCV y HalvingSearchCV para elegir el mejor valor de λ de tal forma que toma en cuenta valores posibles que dan como resultado la solución de mínimos cuadrados y también explora la región en donde se hace la contracción de las estimaciones de los β_j. Para el caso de los datos estandarizados, el caso es muy similar (ver Figuras 6 y (7)).

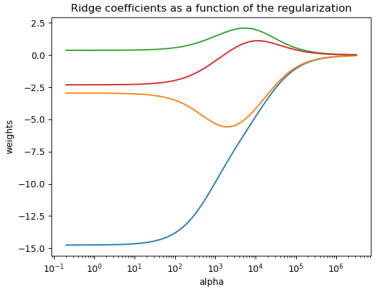

Fig. 6: **Datos estandarizados:** Coeficientes de Lasso en función del valor de λ.

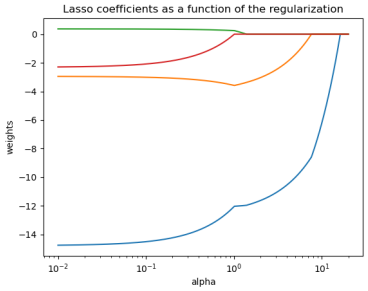

Fig. 7: **Datos estandarizados:** Coeficientes de Lasso en función del valor de λ.

A continuación en las tablas (I y II) se muestran los valores de λ encontrados por c/u de los métodos así como el tiempo de ejecución.

	Regresión Ridge	Regresión Lasso
GridSearchCV	$\lambda = 0.037$	$\lambda = 0.001$
RandomizedSearchCV	$\lambda = 0.031$	$\lambda = 0.001$
HalvingSearchCV	$\lambda = 0.121$	$\lambda = 0.009$

TABLE I: **Datos normalizados:** Valores de λ encontrados en cada método.

	Regresión Ridge	Regresión Lasso
GridSearchCV	$\lambda = 0.758$	$\lambda = 0.01$
RandomizedSearchCV	$\lambda = 0.735$	$\lambda = 0.010$
HalvingSearchCV	$\lambda = 1.576$	$\lambda = 0.080$

TABLE II: **Datos estandarizados:** Valores de λ encontrados en cada método.

Podemos observar que los valores de λ son muy cercanos a cero, esto quiere decir que la solución que más se asemeja a mínimos cuadrados es la que tiene un mejor desempeño. Ahora bien, como los valores de λ son muy similares, los valores de R^2 y MSE dieron resultados prácticamente iguales. En las Figuras (8 y 9) se observan las soluciones de los modelos Ridge (datos normalizados y estandarizados).

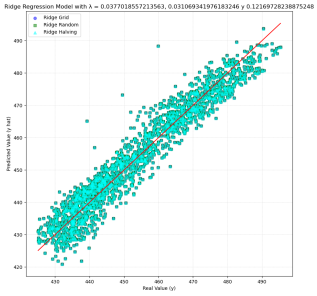

Fig. 8: **Regresores normalizados:** R^2 test = 0.9301 y MSE = 20.2709

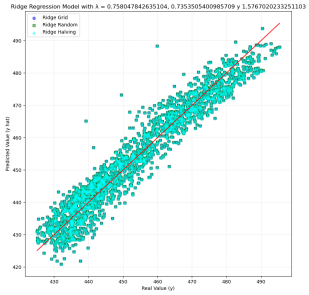

Fig. 9: **Regresores estandarizados:** R^2 test = 0.9301 y MSE test = 20.2719.

Análogamente, para la regresión Lasso se obtuvieron los siguientes resultados (ver Figuras 10 y 11)

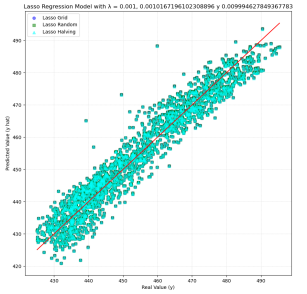

Fig. 10: **Regresores normalizados:** R^2 test = 0.9301 y MSE test = 20.2699.

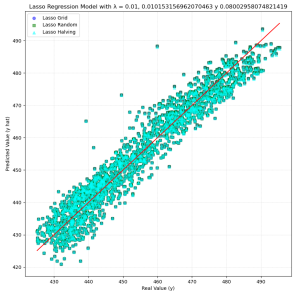

Fig. 11: **Regresores estandarizados:** $R^2 test$ = 0.9301 y MSE test = 20.2663.

Podemos observar que hasta ahora no hay diferencias entre hacer regresión ridge y lasso ya que los valores de penalización son muy cercanos a cero. Ahora bien, a continuación comparamos nuestros resultados con el modelo generado por AutoKeras el cual consiste en una red neuronal. Bajo el esquema en el que los regresores están normalizados, la comparación de la red encontrada por AutoKeras y nuestros modelos Ridge y Lasso se puede apreciar en el Fig. (12)

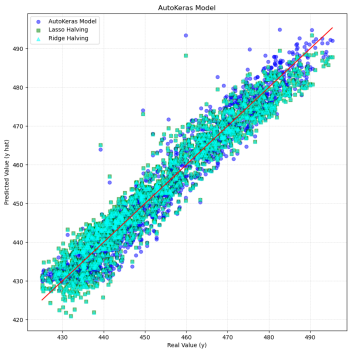

Fig. 12: Comparación del modelo encontrado por AutoKeras y los modelos ajustados de Ridge y Lasso.

El modelo de AutoKeras tuvo las siguientes métricas en el entrenamiento: R^2 train = 0.9373 y MSE train = 18.262. Por otro lado, en el conjunto de prueba se obtuvieron los siguientes valores: R^2 test = 0.9387 y MSE test = 17.7606. Ahora bien, la arquitectura con la que cuenta la red neuronal es la siguiente (ver Fig. (13)):

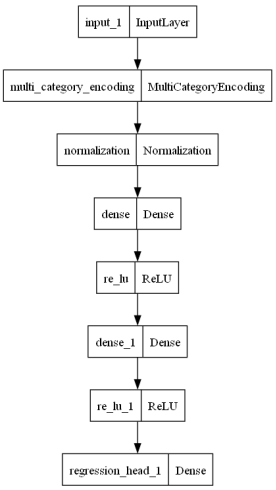

Fig. 13: Arquitectura de la red neuronal.

Del mismo modo, bajo el enfoque en el que los datos están estandarizados, se tiene lo siguiente (ver Fig. 14)

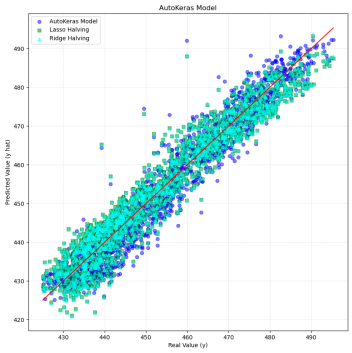

Fig. 14: Comparación del modelo encontrado por AutoKeras y los modelos ajustados de Ridge y Lasso.

El modelo de AutoKeras tuvo las siguientes métricas en el entrenamiento: R^2 train = 0.9336 y MSE train = 19.3318. Por otro lado, en el conjunto de prueba se obtuvieron los siguientes valores: R^2 test = 0.9334 y MSE test = 19.2980. Ahora bien, la arquitectura con la que cuenta la red neuronal es la siguiente (ver Fig. (15)):

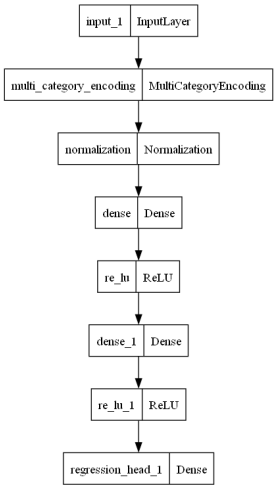

Fig. 15: Arquitectura de la red neuronal.

Ahora bien, en las Figuras (16 y 17) mostramos la graficación de los primeros 100 elementos de los datos de test contra la predicción dada por las redes neuronales 13 y 15:

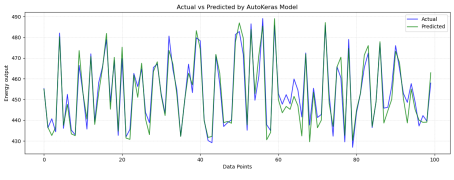

Fig. 16: Predicciones de la arquitectura neuronal encontrada con los datos normalizados.

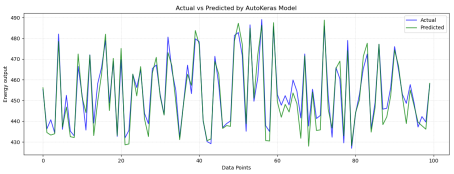

Fig. 17: Predicciones de la arquitectura neuronal encontrada con los datos estandarizados.

En la Fig. (16) se observa que la arquitectura con los datos normalizados tuvo un desempeño ligeramente mayor que la arquitectura con los datos estandarizados.
Finalmente, bajo el enfoque de la libreria de PyCaret, nosotros propusimos la inicialización de los datos con una normalización. Esto nos genero una serie de modelos bajo el mismo enfoque de normalización con sus respectivas métricas de evaluación obtenidas con validación cruzada de 10 folds (ver Fig. 18).

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
xgboost	Extreme Gradient Boosting	2.3694	11.3156	3.3602	0.9603	0.0074	0.0052
lightgbm	Light Gradient Boosting Machine	2.5526	12.1228	3.4759	0.9581	0.0076	0.0056
rf	Random Forest Regressor	2.5241	12.4675	3.5255	0.9569	0.0077	0.0056
et	Extra Trees Regressor	2.5108	12.6813	3.5517	0.9562	0.0078	0.0055
gbr	Gradient Boosting Regressor	3.0171	15.7059	3.9565	0.9457	0.0087	0.0066
knn	K Neighbors Regressor	2.9141	15.9994	3.9960	0.9446	0.0088	0.0064
lar	Least Angle Regression	3.6582	21.2294	4.6030	0.9266	0.0101	0.0081
br	Bayesian Ridge	3.6582	21.2294	4.6030	0.9266	0.0101	0.0081
ridge	Ridge Regression	3.6584	21.2294	4.6030	0.9266	0.0101	0.0081
lr	Linear Regression	3.6582	21.2294	4.6030	0.9266	0.0101	0.0081

Fig. 18: Mejores modelos encontrados con ayuda de PyCaret.

En la Fig. (18) se puede observar que el modelo de regresión lineal y regresión lasso entran en el top 10 de mejores modelos. En este caso por defecto, PyCaret separó los datos en 6697 para entrenamiento y 2871 para prueba. Algunas imágenes informativas que nos da esta librería son las siguientes:

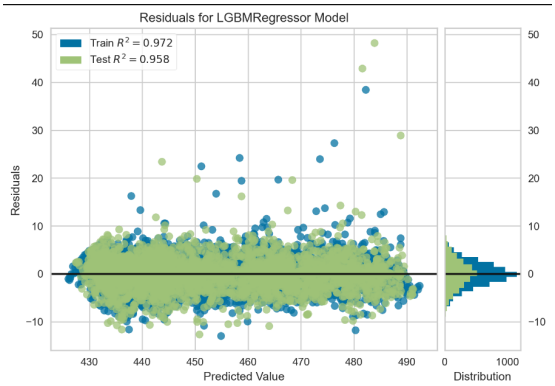

Fig. 19: Residuales y valor de R^2 en test.

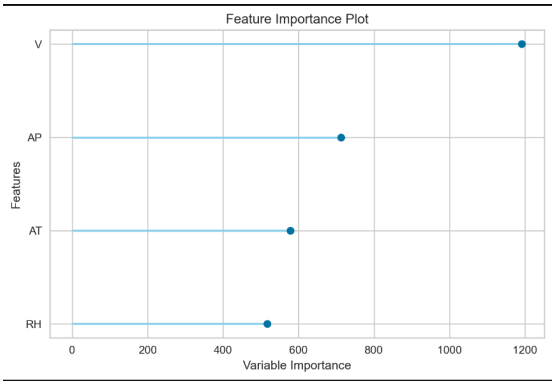

Fig. 20: Vemos que las características más importantes también son las que tienen una mayor correlación lineal con la variable a predecir.

IV. Conclusiones

- Los modelos Lasso y Ridge tienen buenos valores en sus métricas de evaluación (R^2 superior a 0.90).
- La solución óptima de Ridge y Lasso es aquella que no hace contracción de variables.
- AutoKeras es una herramienta de AutoML enfocada en DL y además es fácil de usar.
- Entre los modelos de regresión Ridge y Lasso y las arquitecturas dadas por AutoKeras, las arquitecturas de AutoKeras fueron superiores, sin embargo, a cambio de un modelo mucho más complejo y una mejora despreciable.
- PyCaret es una herramienta sumamente fácil de utilizar y ésta fue la que dio el modelo con los mejores resultados: Extreme Gradient Boosting con los datos normalizados.

V. Bibliografía

References

[1] James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). An introduction to statistical learning: With applications in python. (No Title).
[2] Machine Learning Repository: Combined Cycle Power Plant